

#3
Priority
Paper
2/6/01
Docket No. 1450.1006/HJS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Hideyuki IINO et al.

Group Art Unit:

Serial No.:

Examiner:

Filed: October 4, 2000

For: RESET CONTROL SYSTEM AND METHOD



**SUBMISSION OF CERTIFIED COPY OF PRIOR
FOREIGN APPLICATION IN ACCORDANCE WITH
THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)
herewith a certified copy of the following foreign application(s):

Japanese Patent Application No. 11-315654
Filed: November 5, 1999

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date, as evidenced by the certified papers attached hereto, in accordance with the requirements
of 35 U.S.C. § 119.

Respectfully submitted,
STAAS & HALSEY LLP

Date: October 4, 2000

By: 

H. J. Staas

Registration No. 22,010

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年11月 5日

出 願 番 号

Application Number:

平成11年特許願第315654号

出 願 人

Applicant(s):

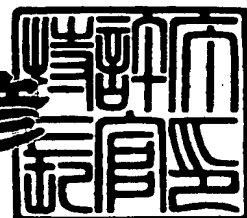
富士通株式会社



2000年 6月29日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3052154

【書類名】 特許願

【整理番号】 9901143

【提出日】 平成11年11月 5日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 1/24

【発明の名称】 リセット制御システムおよび方法

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 飯野 秀之

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 内海 祐之

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 広瀬 佳生

【発明者】

【住所又は居所】 東京都品川区大崎2丁目8番8号 富士通デバイス株式会社内

【氏名】 劉 憲

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100090273

【弁理士】

【氏名又は名称】 國分 孝悦

【電話番号】 03-3590-8901

【手数料の表示】

【予納台帳番号】 035493

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9908504

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 リセット制御システムおよび方法

【特許請求の範囲】

【請求項 1】 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御システムであって、

上記中央演算実行部の機能を別に実現するエミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号とに基づいてシステムリセット信号を生成して出力するシステムリセット出力手段を備え、

上記システムリセット出力手段から出力される上記システムリセット信号を上記中央演算実行部および上記周辺制御部の両チップに供給することを特徴とするリセット制御システム。

【請求項 2】 上記エミュレータの動作時に上記外部リセット信号をマスクするマスク手段を備えることを特徴とする請求項 1 に記載のリセット制御システム。

【請求項 3】 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御システムであって、

上記中央演算実行部の機能を別に実現するエミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号との何れかをシステムリセット信号として選択的に出力するリセット選択手段を備え、

上記リセット選択手段から出力される上記システムリセット信号を上記中央演算実行部および上記周辺制御部の両チップに供給することを特徴とするリセット制御システム。

【請求項 4】 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御方法であって、

中央演算実行部の機能を別に実現するエミュレータの動作時には外部リセット信号にマスクを施し、上記エミュレータのリセット指示に基づくエミュレータリセット信号とマスクの施された外部リセット信号とに基づいてシステムリセット信号を生成して上記中央演算実行部と上記周辺制御部の両チップに供給することを特徴とするリセット制御方法。

【請求項 5】 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御方法であって、

中央演算実行部の機能を別に実現するエミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号との何れかをシステムリセット信号として選択的に出力し、上記システムリセット信号を上記中央演算実行部と上記周辺制御部の両チップに供給することを特徴とするリセット制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、リセット制御システムおよび方法に関し、特に、デバッグサポート機能を内蔵するプロセッサと、それに接続するコンパニオンチップとを備えたシステムのリセット制御方式に用いて好適なものである。

【0002】

【従来の技術】

従来、プロセッサを使用するシステムでは、プログラム開発中のソフトデバッグやシステム開発のハードデバッグを行うために、デバッグサポート機能を有する D S U (Debug Support Unit) をプロセッサに内蔵している。そして、デバッグ用に用意した外部エミュレータをこの D S U に接続し、外部エミュレータからデバッグ用コマンドを D S U に与えることにより、システムデバッグが行われる。

【0003】

図 4 は、プロセッサとエミュレータとからなる従来のシステムの一例を示す図である。

図 4 において、401 はデバッグ処理をエミュレートするエミュレータ、402 は各種データ処理を行うプロセッサである。プロセッサ 402 は、プログラムの命令列を実行するための C P U コア 403 と、デバッグ処理を支援するための D S U 404 とを備えている。D S U 404 は、エミュレータ 401 とのインターフェースを有し、エミュレータ 401 と接続されている。

【0004】

エミュレータ 4 0 1 は、デバッグ用コマンド 4 1 1 を D S U 4 0 4 に送信する。送信するデバッグ用コマンド 4 1 1 は、任意の命令／任意のデータのロード・ストア／任意のデータ値／一命令毎のステップ実行でブレーク割り込みを発生させるブレークコマンド、デバッグ開始時やブレーク割り込み発生後に命令実行開始を指示する実行コマンドがある。また、ブレーク割り込み発生時に内蔵レジスタやメモリの値を確認したり更新したりするためのリード／ライトコマンド、デバッグ処理を最初から行う際などにプロセッサの内部状態を初期化するために使用するリセットコマンドなどがある。

【 0 0 0 5 】

D S U 4 0 4 は、エミュレータ 4 0 1 から送信されたデバッグ用コマンド 4 1 1 を受信すると、デバッグモード信号 4 1 2 を出力し、プロセッサ 4 0 2 を通常動作状態のユーザモードからデバッグ状態のデバッグモードに遷移させる。また、D S U 4 0 4 はデバッグ用コマンドをデコードし、そのデコード結果 4 1 3 を C P U コア 4 0 3 に供給する。

【 0 0 0 6 】

C P U コア 4 0 3 は、供給されたデバッグ用コマンドのデコード結果 4 1 3 に従って処理を実行し、その実行結果 4 1 4 を D S U 4 0 4 に通知する。D S U 4 0 4 は、C P U コア 4 0 3 より受け取った実行結果 4 1 4 を、エミュレータが送信したデバッグ用コマンド 4 1 1 の実行結果 4 1 5 としてエミュレータ 4 0 1 に供給する。

このようにして、エミュレータ 4 0 1 が送信したデバッグ用コマンド 4 1 1 をデバッグサポート機能を有するプロセッサ 4 0 2 が処理実行することにより、システムのデバッグは行われる。

【 0 0 0 7 】

また、最近の高性能プロセッサを利用する P C (Personal Computer) や L B P (Laser Beam Printer) 等のシステムでは、性能および価格等の理由により、用途に特化したシステムを構築するようになってきた。このようなシステムでは、プロセッサにはなるべく C P U コアの機能のみを持たせ、システムの用途に合わせて必要なバス制御部や周辺リソースはプロセッサとは別のコンパニオンチッ



ブに設ける。これにより、異なる用途のシステムで高性能なプロセッサの共通化を図り、システム毎にプロセッサを変更することなく、コンパニオンチップのみ変更して様々なシステムを構築できるようにしている。

【 0 0 0 8 】

図 5 は、プロセッサとコンパニオンチップとからなる従来のシステムの一例を示す図である。

図 5 において、5 0 1 はプロセッサであり、CPU コア 5 0 2 と、プロセッサ 5 0 1 とコンパニオンチップ 5 0 5 とを接続するプロセッサバス 5 0 4 上でのデータ通信を制御するプロセッサバス制御部 5 0 3 とを有する。

【 0 0 0 9 】

5 0 5 はバス制御部や周辺リソースなどを有するコンパニオンチップである。この図 5 の例では、コンパニオンチップ 5 0 5 は、バス制御部としてプロセッサバス 5 0 4 と P C I バス 5 1 2 とのバスプロトコル変換を行うバス変換部 5 0 6 を有する。また、システムに必要な周辺リソースとして、DMAC (Direct Memory Access Controller) 5 0 7、I R C (Interrupt Request Controller) 5 0 8、タイマー部 5 0 9、メモリ制御部 5 1 0、図示しない U A R T (Universal Asynchronous Receiver Transmitter) 等を有する。

【 0 0 1 0 】

ここで、DMAC 5 0 7 は、プロセッサ 5 0 1 を介さずにメモリ 5 1 5 と他の装置との間で直接データを送受信する DMA (Direct Memory Access) 転送を制御するものである。I R C 5 0 8 は、システムに接続されている各装置からの割り込み要求を制御するものであり、タイマー部 5 0 9 はタイムアウト検出等のために時間の計測を行うものである。また、メモリ制御部 5 1 0 は、コンパニオンチップ 5 0 5 の外部に接続された専用メモリ 5 1 1 を制御するものであり、U A R T は、語並列コントローラ等をビット直列通信網に接続するものである。

【 0 0 1 1 】

また、上記 P C I バス 5 1 2 には、B o o t R O M 5 1 3 と I / O 装置 5 1 4 とメモリ 5 1 5 と A S I C (Application Specific Integrated Circuit) 5 1 6 とが接続されており、上記プロセッサ 5 0 1 およびコンパニオンチップ 5 0

5により、これらの動作が制御される。

【0012】

プロセッサ501は、プロセッサバス504を介してコンパニオンチップ505に接続されており、コンパニオンチップ505に内蔵されているDMAC等の周辺リソース507、508、509、510を制御する。また、プロセッサ501がPCIバス512に接続されているリソース513～516にアクセスするときには、プロセッサバス504を介して、コンパニオンチップ505に内蔵されているバス変換部506でプロセッサバス504とPCIバス512とのバスプロトコル変換を行い、PCIバス512に接続されているリソース513～516にアクセスする。

【0013】

次に、図6にDSU内蔵のプロセッサとコンパニオンチップとエミュレータとからなるシステムのデバッグ形態の一例を示す。

図6において、601はエミュレータであり、602はプロセッサである。プロセッサ602は、DSU603とCPUコア604と第1のリセット制御部605とを有する。DSU603は、エミュレータ601とのインタフェースを備えており、エミュレータ601と接続されている。

【0014】

第1のリセット制御部605は、エミュレータ601から出力されたデバッグ用リセットコマンド621をDSU603でデコードして得られる第1のリセット信号625と、図示しない外部のリセット発生回路等から出力された外部リセット信号626からコンパニオンチップ606内の第2のリセット制御部607で生成される第2のリセット信号628との論理和演算を行う。そして、その演算結果を、CPUコア604を初期化する第1の内部リセット信号629としてCPUコア604に出力する。

【0015】

また、コンパニオンチップ606は、上記第2のリセット制御部607と図示しないバス制御部、周辺リソース等を有する。第2のリセット制御部607は、図示しない外部のリセット発生回路等から出力された外部リセット信号626を



第2の内部リセット信号627と第2のリセット信号628とに分配する。第2の内部リセット信号627はコンパニオンチップ606の内部状態を初期化し、第2のリセット信号628はプロセッサ602内の第1のリセット制御部605に出力される。

【0016】

コンパニオンチップ606は、図示しないプロセッサバスおよび個別信号線によりプロセッサ602と接続されており、コンパニオンチップ606内の図示しないバス制御部および周辺リソースは、プロセッサ602により制御されている。

【0017】

次に、図6に示すシステムのデバッグ時の動作について説明する。

エミュレータ601は、デバッグ用コマンド621をDSU603に送信する。デバッグ用コマンド621を受信したDSU603は、デバッグモード信号622を出力し、プロセッサ602を通常動作状態のユーザモードからデバッグ状態のデバッグモードに遷移させる。

【0018】

また、DSU603は、エミュレータ601から供給されたデバッグ用コマンド621をデコードし、その結果、デバッグ用コマンド621がリセットコマンド以外有的时候には、デコード結果をデバッグ指示信号623としてCPUコア604に供給する。

CPUコア604は、供給されたデバッグ指示信号623に従い処理を実行し、実行した結果624をDSU603に通知する。さらに、DSU603は、CPUコア604から受信した実行結果624をエミュレータ601が送信したデバッグ用コマンド621の実行結果630としてエミュレータ601に通知する。

【0019】

一方、エミュレータ601から供給されたデバッグ用コマンド621をDSU603でデコードした結果、リセットコマンドであったときには、DSU603は第1のリセット信号625を第1のリセット制御部605に出力する。



この第1のリセット信号625を受信した第1のリセット制御部605は、外部リセット信号626に基づく第2のリセット信号628との論理和演算を行い、その結果、第1の内部リセット信号629をCPUコア604に出力し、CPUコア604を初期化する。

【0020】

また、図示しない外部のリセット発生回路等により出力された外部リセット信号626は、第2のリセット制御部607に入力される。上記外部リセット信号626を受信した第2のリセット制御部607は、第2の内部リセット信号627を出力してコンパニオンチップ606の内部状態を初期化するとともに、プロセッサ602内の第1のリセット制御部605に第2のリセット信号628を出力する。

【0021】

【発明が解決しようとする課題】

以上のように、最近の高性能なシステムでは、プロセッサとコンパニオンチップとが別チップとして形成されるが、これらは全くの別物ではなく、2チップ合わせてシステム全体としての動作が保証されなければならない。このため、システム開発において、ハードデバッグおよびソフトデバッグを行う際には、プロセッサとコンパニオンチップとの2チップを両方含んだ状態でデバッグを行う必要がある。よって、初期化をする際にも両チップに対して行わないと、正しいデバッグができなくなってしまう。

【0022】

しかしながら、システムの内部状態をリセットする場合、プロセッサ602とコンパニオンチップ606とが別チップとして備えられた図6のような従来のシステムでは、エミュレータ601からのリセット指示により、プロセッサ602を初期化することはできるが、コンパニオンチップ606を初期化することができず、そのためシステム全体としてリセットすることができないという問題があった。

【0023】

また、上記図6のシステムでは、プロセッサ602がユーザモード（通常動作

状態)であるかデバッグモード(デバッグ状態)であるかに関わらず、外部のリセット発生回路等から外部リセット信号626が出力されたときには、プロセッサ602およびコンパニオンチップ606が共に初期化される。このため、エミュレータ601がデバッグ用コマンド621をプロセッサ602に送信することにより、プロセッサ602がデバッグモードに遷移してシステムのデバッグを行っているときに、リセット発生回路の不具合などの何らかの原因で外部リセット信号626が不意に inputs されると、プロセッサ602およびコンパニオンチップ606が初期化され、CPUコア604の内部等に記憶しているデバッグ中の内部状態が消失してしまうという問題があった。

【0024】

本発明は、このような問題を解決するために成されたものであり、エミュレータからのリセット指示により、プロセッサおよびコンパニオンチップの2チップともに初期化できるようにすることを目的とする。

また、本発明は、プロセッサがデバッグモードにあるときは、外部リセット信号が入力されても、初期化されることなくデバッグ中の内部状態を維持できるようにすることをも目的とする。

【0025】

【課題を解決するための手段】

本発明によるリセット制御システムは、中央演算実行部と周辺制御部とが別チップ上に形成されたシステムにおいて、エミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号とに基づいてシステムリセット信号を生成して上記中央演算実行部および上記周辺制御部の両チップに供給する。

【0026】

また、本発明によるリセット制御システムの他の特徴とするところは、エミュレータの動作時に外部リセット信号をマスクするマスク手段を備える。

【0027】

また、中央演算実行部と周辺制御部のうち少なくとも一方のチップに、リセット後の起動タイミングを上記中央演算実行部と上記周辺制御部とで同期させる同期手段を備えるようにしても良い。



【 0 0 2 8 】

また、エミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号とに基づいてシステムリセット信号を生成して出力するシステムリセット出力手段を中央演算実行部のチップ内に備えるようにしても良い。

【 0 0 2 9 】

上記のように構成した本発明によれば、エミュレータからのリセット指示に基づくエミュレータリセット信号と外部のリセット発生回路等からの外部リセット信号とに基づいて生成されたシステムリセット信号が、中央演算実行部ばかりでなく、これとは別チップの周辺制御部にも供給されることとなるので、エミュレータリセット信号により上記中央演算実行部と上記周辺制御部との2チップとも初期化することができるようになる。

【 0 0 3 0 】

また、本発明の他の特徴によれば、エミュレータを動作させているときには、外部リセット信号がマスクされて無効にされるので、デバッグ中に不意に外部リセット信号が入力されても、当該外部リセット信号によって意図に反してリセットが行われてしまう不都合を回避することができるようになる。

【 0 0 3 1 】

また、同期手段を設けた場合は、中央演算実行部と周辺制御部との内部回路の違い等により生じるリセット解除の時間差が調整され、リセット動作後に上記中央演算実行部と上記周辺制御部とが同期して起動されることとなるので、先に起動した方のチップからまだ起動していないチップへアクセスしてしまう不都合が防止できるようになる。

【 0 0 3 2 】

また、中央演算実行部にシステムリセット出力手段を設けた場合は、エミュレータから中央演算実行部に対するリセット指示はコマンドの形態で行うことが可能となり、リセットの種類を増やすときでも、エミュレータとのインタフェースを変更することなくコマンド拡張のみでリセット指示の種類を増やすことができるようになる。また、外部リセット信号は中央演算実行部に直接入力されるので、エミュレータの接続、未接続に関わらず外部リセット信号の入力経路を変更す

る必要がないので、通常使用の形態と同じ状態でデバッグを行うことができるようになる。

【 0 0 3 3 】

【発明の実施の形態】

以下に、本発明の一実施形態を図面に基づいて説明する。

図 1 は、第 1 の実施形態によるリセット制御システムを適用したシステムの一構成例を示すブロック図である。このシステムは、プロセッサとこれに接続するコンパニオンチップとを備えたものである。

なお、本実施形態ではリセットの種類として、例えば、パワーオンリセットとハードリセットの場合について説明する。パワーオンリセットは、電源投入時のリセットであり、プロセッサおよびコンパニオンチップの全内部状態を初期化するものである。一方、ハードリセットはプロセッサおよびコンパニオンチップの一部の内部状態を初期化するものである。

【 0 0 3 4 】

図 1 において、10 はプロセッサ、20 はコンパニオンチップ、30 はエミュレータである。プロセッサ 10 は、DSU 11 と CPU コア部 12 と第 1 のリセット制御部 13 とを備えている。DSU 11 は、エミュレータ 30 とのインタフェースを有し、エミュレータ 30 から送信されるデバッグ用コマンド 101 を受信し、その実行結果 102 をエミュレータ 30 に送信する。DSU 11 は、エミュレータ 30 から送信されるデバッグ用コマンド 101 を受信すると、デバッグモード信号 103 を出力し、プロセッサ 10 をユーザモードからデバッグモードに遷移させる。

【 0 0 3 5 】

また、DSU 11 は、受信したデバッグ用コマンド 101 をデコードし、デコード結果をデバッグ指示信号 104 として CPU コア部 12 に供給する。CPU コア部 12 は、供給されたデバッグ指示信号 104 に従い処理を実行し、その実行結果 105 を DSU 11 に通知する。さらに、DSU 11 は、CPU コア部 12 から受信した実行結果 105 をエミュレータ 30 が送信したデバッグ用コマンド 101 の実行結果 102 としてエミュレータ 30 に送信する。

【 0 0 3 6 】

CPUコア部 1 2 は、CPUコアとプロセッサバス制御部とを含み、図示しないプロセッサバス等を介してコンパニオンチップ 2 0 と接続されている。

第 1 のリセット制御部 1 3 は、エミュレータ 3 0 からパワーオンリセットに対応した第 1 のシステムリセット信号 1 0 9 が入力されると、第 1 のプロセッサ内部リセット信号 1 1 1 を生成し、CPUコア部 1 2 に出力する。また、ハードリセットに対応した第 2 のシステムリセット信号 1 1 0 が入力されたときには、第 2 のプロセッサ内部リセット信号 1 1 2 を生成し、CPUコア部 1 2 に出力する。

【 0 0 3 7 】

コンパニオンチップ 2 0 は、第 2 のリセット制御部 2 1 と図示しないバス変換部や周辺リソース等で構成されている。

第 2 のリセット制御部 2 1 は、第 1 のリセット制御部 1 3 と同様に、第 1 のシステムリセット信号 1 0 9 が入力されると、第 1 のコンパニオンチップ内部リセット信号 1 1 3 を生成し、コンパニオンチップ 2 0 内のバス変換部や周辺リソース等に出力する。また、第 2 のシステムリセット信号 1 1 0 が入力されたときには、第 2 のコンパニオンチップ内部リセット信号 1 1 4 を生成し、コンパニオンチップ 2 0 内のバス変換部や周辺リソース等に出力する。

【 0 0 3 8 】

エミュレータ 3 0 は、エミュレータリセット信号 1 0 6、1 0 7 およびマスク信号 1 0 8 等を生成する信号生成部 3 1 と、リセット信号の制御を行う第 3 のリセット制御部 3 2 とを含む。また、エミュレータ 3 0 は、リセットコマンドを除くデバッグ用コマンド 1 0 1 を DSU 1 1 に送信し、その実行結果 1 0 2 を DSU 1 1 から受信する。

【 0 0 3 9 】

信号生成部 3 1 は、エミュレータ 3 0 がデバッグ用コマンド 1 0 1 を DSU 1 1 に送信したとき、すなわちデバッグを行っているときにはマスク信号 1 0 8 を第 3 のリセット制御部 3 2 に出力する。また、プロセッサ 1 0 およびコンパニオンチップ 2 0 のリセット要求時には、パワーオンリセットあるいはハードリセッ

ト等のリセット要求に対応したエミュレータリセット信号 1 0 6、1 0 7 を生成し第 3 のリセット制御部 3 2 に出力する。

【 0 0 4 0 】

第 3 のリセット制御部 3 2 は、マスク回路 3 3 と O R 回路 3 4、3 5 とを含む。マスク回路 3 3 は、信号生成部 3 1 から入力されるマスク信号 1 0 8 に従い、図示しない外部のリセット発生回路等から入力されるパワーオンリセットあるいはハードリセットに対応した外部リセット信号 1 1 5、1 1 6 をマスクする。このマスク回路 3 3 は本発明のマスク手段を構成する。

【 0 0 4 1 】

O R 回路 3 4、3 5 は、信号生成部 3 1 から出力されるエミュレータリセット信号 1 0 6、1 0 7 と、マスク回路 3 3 から出力される外部リセット信号との論理和演算を行い、その結果をシステムリセット信号 1 0 9、1 1 0 として出力する。この O R 回路 3 4、3 5 は本発明のシステムリセット出力手段を構成する。

【 0 0 4 2 】

次に、図 1 に示すリセット制御システムの動作について説明する。

エミュレータ 3 0 からデバッグ用コマンド 1 0 1 が送信されておらず、また、エミュレータ 3 0 からリセット要求も出されていない状態、すなわち、プロセッサ 1 0 がユーザモードで通常に動作している状態であるとする。このとき、エミュレータ 3 0 内の信号生成部 3 1 からは、第 1 のエミュレータリセット信号 1 0 6、第 2 のエミュレータリセット信号 1 0 7、マスク信号 1 0 8 の何れも出力されない。

【 0 0 4 3 】

このユーザモードにおいて、例えば、図示しない外部リセット発生回路等によりパワーオンリセットに応じた第 1 の外部リセット信号 1 1 5 が発生されると、発生された第 1 の外部リセット信号 1 1 5 はエミュレータ 3 0 内のマスク回路 3 3 に入力される。マスク回路 3 3 では、信号生成部 3 1 からマスク信号 1 0 8 が出力されていないため、入力された第 1 の外部リセット信号 1 1 5 をマスクせずに、O R 回路 3 4 に出力する。

【 0 0 4 4 】

○ R 回路 3 4 では、信号生成部 3 1 から出力される第 1 のエミュレータリセット信号 1 0 6 とマスク回路 3 3 から出力される第 1 の外部リセット信号との論理和演算を行い、その演算結果を、第 1 のシステムリセット信号 1 0 9 として出力する。この場合、第 1 のエミュレータ信号 1 0 6 は出力されていないので、マスク回路 3 3 からの信号がそのまま○ R 回路 3 4 を通過する。○ R 回路 3 4 から出力された第 1 のシステムリセット信号 1 0 9 は、分配されてプロセッサ 1 0 内の第 1 のリセット制御部 1 3 と、コンパニオンチップ 2 0 内の第 2 のリセット制御部 2 1 に供給される。

【 0 0 4 5 】

なお、ユーザモード時において、ハードリセットに応じた第 2 の外部リセット信号 1 1 6 が発生した場合も、上記第 1 の外部リセット信号 1 1 5 が発生した場合と同様に、当該発生した第 2 の外部リセット信号 1 1 6 がマスク回路 3 3 および○ R 回路 3 5 を通過して、第 2 のシステムリセット信号 1 1 0 としてプロセッサ 1 0 内の第 1 のリセット制御部 1 3 と、コンパニオンチップ 2 0 内の第 2 のリセット制御部 2 1 に供給される。

【 0 0 4 6 】

一方、プロセッサ 1 0 がデバッグモードで動作しているとき、信号生成部 3 1 はマスク信号 1 0 8 を出力している。

また、プロセッサ 1 0 がデバッグモードで動作中に、プロセッサ 1 0 およびコンパニオンチップ 2 0 のリセット要求がエミュレータ 3 0 で行われると、エミュレータ 3 0 内の信号生成部 3 1 はリセット要求に対応したエミュレータリセット信号 1 0 6、1 0 7 を出力する。リセット要求がパワーオンリセットであった場合は第 1 のエミュレータリセット信号 1 0 6 を出力し、ハードリセットであった場合は第 2 のエミュレータリセット信号 1 0 7 を出力する。

【 0 0 4 7 】

以下、エミュレータ 3 0 へのリセット要求が、例えばパワーオンリセットである場合について説明する。

エミュレータ 3 0 でパワーオンリセットのリセット要求を受けると、信号生成部 3 1 は、第 1 のエミュレータリセット信号 1 0 6 を○ R 回路 3 4 に出力する。

【 0 0 4 8 】

OR回路 3 4 では、第 1 のエミュレータリセット信号 1 0 6 とマスク回路 3 3 から出力される第 1 の外部リセット信号との論理和演算が行われ、その演算結果が第 1 のシステムリセット信号 1 0 9 として出力される。この場合、仮に何らかの原因で第 1 の外部リセット信号 1 1 5 が発生しても、マスク回路 3 3 において、信号生成部 3 1 から出力されているマスク信号 1 0 8 に従ってマスクが施されているため、第 1 の外部リセット信号 1 1 5 は OR 回路 3 4 には出力されない。

【 0 0 4 9 】

したがって、信号生成部 3 1 から出力された第 1 のエミュレータリセット信号 1 0 6 が OR 回路 3 4 を通過し、第 1 のシステムリセット信号 1 0 9 として出力される。OR 回路 3 4 から出力された第 1 のシステムリセット信号 1 0 9 は、分配されてプロセッサ 1 0 内の第 1 のリセット制御部 1 3 と、コンパニオンチップ 2 0 内の第 2 のリセット制御部 2 1 に供給される。

【 0 0 5 0 】

また、何らかの原因で第 2 の外部リセット信号 1 1 6 が発生したときも、マスク回路 3 3 において、信号生成部 3 1 から出力されているマスク信号 1 0 8 に従ってマスクが施されているため、第 2 の外部リセット信号 1 1 6 も OR 回路 3 5 に出力されることはない。したがって、上記 OR 回路 3 4 を通過した第 1 のエミュレータリセット信号 1 0 6 のみが、第 1 のシステムリセット信号 1 0 9 としてプロセッサ 1 0 内の第 1 のリセット制御部 1 3 と、コンパニオンチップ 2 0 内の第 2 のリセット制御部 2 1 に供給される。

【 0 0 5 1 】

プロセッサ 1 0 内の第 1 のリセット制御部 1 3 は、供給された第 1 のシステムリセット信号 1 0 9 に従い、第 1 のプロセッサ内部リセット信号 1 1 1 を生成し、CPU コア部 1 2 を初期化する。また、コンパニオンチップ 2 0 内の第 2 のリセット制御部 2 1 も同様に、第 1 のコンパニオンチップ内部リセット信号 1 1 3 を生成し、コンパニオンチップ 2 0 内のバス変換部や周辺リソース等を初期化する。

【 0 0 5 2 】



また、プロセッサ 1 0 がデバッグモードで動作中に、エミュレータ 3 0 からデバッグ用コマンド 1 0 1 が D S U 1 1 に送信され、送信されたデバッグ用コマンド 1 0 1 に基づき、C P U コア部 1 2 が処理を実行しているとする。このとき、仮に何らかの原因で、図示しない外部リセット発生回路等により外部リセット信号 1 1 5 または 1 1 6 が発生されても、この発生した外部リセット信号 1 1 5、1 1 6 は、エミュレータ 3 0 内のマスク回路 3 3 において、信号生成部 3 1 から出力されているマスク信号 1 0 8 に従ってマスクされ、無効にされる。

【 0 0 5 3 】

したがって、エミュレータ 3 0 からのデバッグ用コマンド 1 0 1 に基づいて C P U コア部 1 2 が処理を実行しているときには、図示しない外部リセット発生回路等により発生された外部リセット信号 1 1 5 または 1 1 6 によって、C P U コア部 1 2 は内部状態を初期化されることなく処理を継続することができる。

【 0 0 5 4 】

以上のように、本実施形態によれば、エミュレータ 3 0 内の信号生成部 3 1 から入力されるエミュレータリセット信号 1 0 6、1 0 7 と、外部のリセット発生回路等から入力される外部リセット信号 1 1 5、1 1 6 とを O R 回路 3 4、3 5 で論理和演算を行い、その演算結果を、システムリセット信号 1 0 9、1 1 0 として出力し、プロセッサ 1 0 およびコンパニオンチップ 2 0 に分配供給するようにする。これにより、通常動作状態のときだけでなく、システムのデバッグを行っているときでも、エミュレータ 3 0 からのリセット指示により生成されるエミュレータリセット信号 1 0 6、1 0 7 により、プロセッサ 1 0 とコンパニオンチップ 2 0 との 2 チップとも初期化することができるようになる。

【 0 0 5 5 】

また、プロセッサ 1 0 がデバッグモードで動作しているときには、外部リセット信号 1 1 5、1 1 6 をマスクするマスク信号 1 0 8 を、信号生成部 3 1 からマスク回路 3 3 に出力するようにする。これにより、システムのデバッグを行っているときには、外部のリセット発生回路の不具合等により不意に入力された外部リセット信号 1 1 5、1 1 6 を、マスク回路 3 3 によってマスクすることができ、デバッグ中に外部リセット信号 1 1 5、1 1 6 によりプロセッサ 1 0 およびコ

ンパニオンチップ 2 0 が初期化されることを防ぐことができるようになる。

【 0 0 5 6 】

なお、本実施形態ではプロセッサ 1 0 内に第 1 のリセット制御部 1 3 を設け、コンパニオンチップ 2 0 内に第 2 のリセット制御部 2 1 を設けたが、リセット制御部 1 3、2 1 を設けず、第 1 および第 2 のシステムリセット信号 1 0 9、1 1 0 を CPU コア部 1 2、コンパニオンチップ 2 0 内のバス変換部や周辺リソース等に直接入力しても良い。

【 0 0 5 7 】

次に、本発明の第 2 の実施形態について説明する。

図 2 は、第 2 の実施形態によるリセット制御システムを適用したシステムの構成例を示すブロック図である。なお、この図 2 において、図 1 に示したブロックと同じブロックには同一の符号を付し、重複する説明は省略する。

【 0 0 5 8 】

図 1 に示した第 1 の実施形態では、エミュレータ 3 0 内にシステムリセット信号を生成するリセット制御部を設けていたが、本実施形態ではプロセッサ 4 0 内にシステムリセット信号を生成するリセット制御部を設けている。

また、図 1 に示した第 1 の実施形態では、プロセッサ 1 0 内に第 1 のリセット制御部 1 3 を設け、コンパニオンチップ 2 0 内に第 2 のリセット制御部 2 1 を設けていたが、本実施形態では、第 1 および第 2 のリセット制御部を備えない形態について説明する。

【 0 0 5 9 】

図 2 において、4 0 はプロセッサ、5 0 はコンパニオンチップ、6 0 はエミュレータである。プロセッサ 4 0 は、DSU 4 1 と CPU コア部 1 2 とリセット制御部 4 2 とを備えている。DSU 4 1 は、エミュレータ 6 0 とのインタフェースを有し、エミュレータ 6 0 から送信されるデバッグ用コマンド 1 0 1 を受信し、その実行結果 1 0 2 をエミュレータ 6 0 に送信する。DSU 4 1 は、エミュレータ 6 0 から送信されるデバッグ用コマンド 1 0 1 を受信すると、マスク信号 1 1 9 およびデバッグモード信号 1 0 3 を出力し、プロセッサ 4 0 をユーザモードからデバッグモードに遷移させる。また、受信したデバッグ用コマンド 1 0 1 をデ

コードする。

【0060】

上記デコードの結果、エミュレータ60から受信したデバッグ用コマンド101がパワーオンリセットのリセットコマンドであったときには、パワーオンリセットに応じた第1のエミュレータリセット信号117を生成し、OR回路44に出力する。また、受信したデバッグ用コマンド101がハードリセットのリセットコマンドであったときには、ハードリセットに応じた第2のエミュレータリセット信号118を生成し、OR回路45に出力する。

【0061】

一方、デコードの結果、エミュレータ60から受信したデバッグ用コマンド101がリセットコマンド以外であったときには、デコード結果をデバッグ指示信号104としてCPUコア部12に供給する。CPUコア部12は、供給されたデバッグ指示信号104に従い処理を実行し、その実行結果105をDSU41に通知する。さらに、DSU41は、CPUコア部12から受信した実行結果105をエミュレータ60が送信したデバッグ用コマンド101の実行結果102としてエミュレータ60に送信する。

【0062】

上記リセット制御部42は、図示しない外部のリセット発生回路等から入力されるパワーオンリセットに応じた第1の外部リセット信号115およびハードリセットに応じた第2の外部リセット信号116と、DSU41から入力される第1および第2のエミュレータリセット信号117、118とを制御して第1および第2のシステムリセット信号120、121を出力する。

以下に、このリセット制御部42内の構成を説明する。

【0063】

マスク回路43は、DSU41から入力されるマスク信号119に従い、第1および第2の外部リセット信号115、116をマスクする。このマスク回路43は、本発明のマスク手段を構成する。

【0064】

OR回路44、45は、DSU41から出力される第1および第2のエミュレ

ータリセット信号 1 1 7、1 1 8 と、マスク回路 4 3 から出力される第 1 および第 2 の外部リセット信号との論理和演算をそれぞれ行い、その結果を第 1 および第 2 のシステムリセット信号 1 2 0、1 2 1 として出力する。この O R 回路 4 4、4 5 は本発明のシステムリセット出力手段を構成する。

【 0 0 6 5 】

4 6、4 7 はカウンタであり、プロセッサ 4 0 とコンパニオンチップ 5 0 との内部回路の違い等によるリセット完了時間の差を調整するためのものである。カウンタ 4 6、4 7 は、プロセッサ 4 0 のリセット解除からコンパニオンチップ 5 0 のリセット解除までに相当する時間を内部クロックを用いてカウントする。そして、プロセッサ 4 0 のリセット解除からコンパニオンチップ 5 0 のリセット解除までに相当する時間をカウントした後、O R 回路 4 8 に起動信号を出力する。

【 0 0 6 6 】

すなわち、カウンタ 4 6、4 7 は、システムリセット信号 1 2 0、1 2 1 のネゲートを検出してから一定時間を経過した後、O R 回路 4 8 に起動信号を出力する。O R 回路 4 8 は、カウンタ 4 6、4 7 がそれぞれ出力する起動信号の論理和演算を行い、その結果を起動許可信号 1 2 2 として C P U コア部 1 2 に出力することにより、C P U コア部 1 2 を起動する。このカウンタ 4 6、4 7 および O R 回路 4 8 は、リセットネゲート後のプロセッサ 4 0 とコンパニオンチップ 5 0 との起動タイミングの同期をとるブロックであり、本発明の同期手段を構成する。

【 0 0 6 7 】

コンパニオンチップ 5 0 は、図示しないバス変換部や周辺リソース等で構成されており、プロセッサ 4 0 とプロセッサバス等で接続されている。

なお、本実施形態はコンパニオンチップ 5 0 にリセット制御部を具備しない形態を示しているが、プロセッサ 4 0 から入力されるシステムリセット信号 1 2 0、1 2 1 から内部リセット信号を生成するリセット制御部をコンパニオンチップ 5 0 内に具備しても良い。

【 0 0 6 8 】

次に、図 2 に示すリセット制御システムの動作について説明する。

エミュレータ 6 0 からデバッグ用コマンド 1 0 1 が送信されておらず、プロセ



ッサ 4 0 がユーザモードで通常に動作しているとする。このとき、DSU 4 1 からは第 1 のエミュレータリセット信号 1 1 7、第 2 のエミュレータリセット信号 1 1 8、マスク信号 1 1 9 の何れも出力されない。

【 0 0 6 9 】

このユーザモードにおいて、例えば、図示しない外部リセット発生回路等によりパワーオンリセットに応じた第 1 の外部リセット信号 1 1 5 が発生されると、発生された第 1 の外部リセット信号 1 1 5 はプロセッサ 4 0 内のマスク回路 4 3 に入力される。マスク回路 4 3 では、DSU 4 1 からマスク信号 1 1 9 が出力されていないため、入力された第 1 の外部リセット信号 1 1 5 をマスクせずに、OR 回路 4 4 に出力する。

【 0 0 7 0 】

OR 回路 4 4 では、DSU 4 1 から出力される第 1 のエミュレータリセット信号 1 1 7 とマスク回路 4 3 から出力される第 1 の外部リセット信号との論理和演算を行い、その演算結果を、第 1 のシステムリセット信号 1 2 0 として出力する。この場合、第 1 のエミュレータ信号 1 1 7 は出力されていないので、マスク回路 4 3 からの信号がそのまま OR 回路 4 4 を通過する。OR 回路 4 4 から出力された第 1 のシステムリセット信号 1 2 0 は、分配されて CPU コア部 1 2、コンパニオンチップ 5 0、カウンタ 4 6 に供給される。

【 0 0 7 1 】

CPU コア部 1 2 およびコンパニオンチップ 5 0 に供給された第 1 のシステムリセット信号 1 2 0 に従い、CPU コア部 1 2 およびコンパニオンチップ 5 0 が初期化される。また、カウンタ 4 6 は、供給された第 1 のシステムリセット信号 1 2 0 のネゲートを検出し、リセットネゲート検出から一定時間をカウントした後、OR 回路 4 8 に起動信号を出力する。

【 0 0 7 2 】

OR 回路 4 8 は、カウンタ 4 6、4 7 のそれぞれから出力される起動信号の論理和演算を行い、その結果を起動許可信号 1 2 2 として CPU コア部 1 2 に供給する。この場合、カウンタ 4 7 から起動信号は出力されていないので、カウンタ 4 6 の出力が OR 回路 4 8 をそのまま通過する。CPU コア部 1 2 はリセット処



理の後、この起動許可信号 1 2 2 を受信して動作を開始し、リセットベクターから処理を実行する。

【 0 0 7 3 】

なお、ハードリセットに応じた第 2 の外部リセット信号 1 1 6 が入力された場合は、第 1 の外部リセット信号 1 1 5 が入力された場合とはリセット範囲が異なるのみで同様の動作である。

【 0 0 7 4 】

一方、エミュレータ 6 0 からデバッグ用コマンド 1 0 1 が D S U 4 1 に送信されて、プロセッサ 4 0 がデバッグモードで動作しているとき、D S U 4 1 はマスク信号 1 1 9 とを出力している。

【 0 0 7 5 】

さらに、エミュレータ 6 0 から送信されたデバッグ用コマンド 1 0 1 がパワーオンリセットあるいはハードリセットのリセットコマンドであったときには、D S U 4 1 は受信したリセットコマンドに対応した第 1 のエミュレータリセット信号 1 1 7 または第 2 のエミュレータリセット信号 1 1 8 の何れかの信号を出力する。

以下では、受信したデバッグ用コマンド 1 0 1 がパワーオンリセットのリセットコマンドであり、上記リセットコマンドに対応した第 1 のエミュレータリセット信号 1 1 7 が出力された場合について説明する。

【 0 0 7 6 】

D S U 4 1 が、エミュレータ 6 0 からパワーオンリセットのリセットコマンドを受信すると、D S U 4 1 は第 1 のエミュレータリセット信号 1 1 7 を O R 回路 4 4 に出力する。O R 回路 4 4 では、D S U 4 1 から出力される第 1 のエミュレータリセット信号 1 1 7 と、マスク回路 4 3 から出力される第 1 の外部リセット信号との論理和演算を行い、その演算結果を、第 1 のシステムリセット信号 1 2 0 として出力する。この場合、第 1 の外部リセット信号 1 1 5 は、マスク回路 4 3 において、D S U 4 1 から出力されているマスク信号 1 1 9 に従ってマスクされているため、O R 回路 4 4 には出力されない。

【 0 0 7 7 】



したがって、DSU41から出力された第1のエミュレータリセット信号117がOR回路44を通過し、第1のシステムリセット信号120として出力される。OR回路44から出力された第1のシステムリセット信号120は、分配されて、CPUコア部12、コンパニオンチップ50、カウンタ46に供給される。

【0078】

CPUコア部12およびコンパニオンチップ50に供給された第1のシステムリセット信号120に従い、CPUコア部12およびコンパニオンチップ50が初期化される。また、カウンタ46は、供給された第1のシステムリセット信号120のネゲートを検出し、リセットネゲート検出から一定時間を経過した後、OR回路48に起動信号を出力する。

【0079】

OR回路48は、カウンタ46、47のそれぞれから出力される起動信号の論理和演算を行い、その結果を起動許可信号122としてCPUコア部12に供給する。この場合、カウンタ47から起動信号は出力されていないので、カウンタ46の出力がOR回路48からそのまま通過する。CPUコア部12はリセット処理の後、この起動許可信号122を受信して動作を開始し、リセット直後に読み出すべきプログラムの先頭アドレスであるリセットベクターから処理を実行する。

【0080】

また、何らかの原因で第2の外部リセット信号116が発生したときも、マスク回路43において、DSU41から出力されているマスク信号119に従ってマスクが施されているため、第2の外部リセット信号116もOR回路45に出力されることはない。したがって、上記OR回路44を通過した第1のエミュレータリセット信号117のみが、第1のシステムリセット信号120としてCPUコア部12、コンパニオンチップ50、カウンタ46に供給される。

【0081】

また、プロセッサ40がデバッグモードで動作中に、エミュレータ60からリセットコマンド以外のデバッグ用コマンド101がDSU41に送信され、送信

されたデバッグ用コマンド 1 0 1 に基づき、CPU コア部 1 2 が処理を実行しているとする。このとき、仮に何らかの原因で、図示しない外部リセット発生回路等により外部リセット信号 1 1 5 または 1 1 6 が発生されても、この発生した外部リセット信号 1 1 5、1 1 6 は、プロセッサ 4 0 内のマスク回路 4 3 において、DSU 4 1 から出力されているマスク信号 1 1 9 に従ってマスクされ、無効にされる。

【0 0 8 2】

したがって、エミュレータ 6 0 からのリセットコマンド以外のデバッグ用コマンド 1 0 1 に基づいて CPU コア部 1 2 が処理を実行しているときには、図示しない外部リセット発生回路等により発生された外部リセット信号 1 1 5 または 1 1 6 によって、CPU コア部 1 2 は内部状態を初期化されることなく処理を継続することができる。

【0 0 8 3】

以上のように、第 2 の実施形態によれば、外部のリセット発生回路等から入力される外部リセット信号 1 1 5、1 1 6 と、DSU 4 1 から入力されるエミュレータリセット信号 1 1 7、1 1 8 とを OR 回路 4 4、4 5 で論理和演算を行い、その演算結果を、システムリセット信号 1 2 0、1 2 1 として出力し、CPU コア部 1 2 およびコンパニオンチップ 5 0 に分配し供給するようにする。これにより、通常動作状態のときだけでなく、システムのデバッグを行っているときでも、エミュレータ 6 0 からのリセット指示により生成されるエミュレータリセット信号 1 1 7、1 1 8 により、プロセッサ 4 0 とコンパニオンチップ 5 0 との 2 チップとも初期化することができるようになる。

【0 0 8 4】

また、プロセッサ 4 0 がデバッグモードで動作しているときには、外部リセット信号 1 1 5、1 1 6 をマスクするマスク信号 1 1 9 を、DSU 4 1 からマスク回路 4 3 に出力するようにする。これにより、システムのデバッグを行っているときには、外部のリセット発生回路の不具合等により不意に入力された外部リセット信号 1 1 5、1 1 6 を、マスク回路 4 3 によってマスクすることができ、デバッグ中に外部リセット信号 1 1 5、1 1 6 によりプロセッサ 4 0 およびコンパ

ニオンチップ 5 0 が初期化されることを防ぐことができるようになる。

【 0 0 8 5 】

さらに、本実施形態によれば、エミュレータ内の O R 回路を用いて生成したシステムリセット信号をプロセッサに供給する第 1 の実施形態と異なり、O R 回路 4 4、4 5 を備えたりセット制御部 4 2 をプロセッサ 4 0 内に設けてシステムリセット信号 1 2 0、1 2 1 をプロセッサ 4 0 内で生成するようにしたので、エミュレータ 6 0 からプロセッサ 4 0 へのリセット要求をコマンドで行うことが可能となり、外部リセット信号の種類が増え、それに伴いエミュレータ 6 0 からのリセット指示の種類を増やすときでも、エミュレータ 6 0 とプロセッサ 4 0 のインタフェースを変更することなく、エミュレータ 6 0 が送信するデバッグ用コマンド 1 0 1 のビット配列を変更するなどしてコマンドの拡張を行うことにより、リセット信号の種類を容易に増加させることができる。

【 0 0 8 6 】

また、本実施形態では、外部リセット信号 1 1 5、1 1 6 とエミュレータリセット信号 1 1 7、1 1 8 からシステムリセット信号 1 2 0、1 2 1 を生成するリセット制御部 4 2 をプロセッサ 4 0 内に設け、外部リセット信号 1 1 5、1 1 6 をプロセッサ 4 0 に直接入力するようにしている。これにより、エミュレータ 6 0 の接続、未接続に関わらず外部リセット信号 1 1 5、1 1 6 の入力経路を変更する必要がなく、通常使用の形態と同じ状態でデバッグを行うことができる。また、通常に使用していてソフトウェアの更新等によりデバッグを行う機会が発生しても容易にデバッグを行うことができる。

【 0 0 8 7 】

また、本実施形態では、カウンタ 4 6、4 7 を設け、システムリセット信号 1 2 0、1 2 1 のネゲートを検出してから一定時間が経過した後、C P U コア部 1 2 の起動を許可する起動許可信号 1 2 2 を C P U コア部 1 2 に出力するようにしている。これにより、プロセッサ 4 0 とコンパニオンチップ 5 0 との内部回路の違い等によるリセット解除の時間差を調整し、プロセッサ 4 0 とコンパニオンチップ 5 0 とを同期して起動させることができる。すなわち、プロセッサ 4 0 とコンパニオンチップ 5 0 とのリセット後の起動までの時間差を調整することができ

、リセット解除後の動作を保証することができる。例えば、先にリセット解除されたプロセッサ 4 0 がコンパニオンチップ 5 0 のリセット動作中にコンパニオンチップ 5 0 にアクセスすることで、コンパニオンチップ 5 0 に P C I バス等を介して接続された周辺リソースにアクセスできない不都合を防ぐことができる。

【 0 0 8 8 】

なお、本実施形態では D S U 4 1 がエミュレータ 6 0 からデバッグ用コマンド 1 0 1 を受信したとき、外部リセット信号 1 1 5、1 1 6 をマスクするためマスク信号 1 1 9 を生成しているが、マスク信号 1 1 9 の代わりに D S U 4 1 がプロセッサ 4 0 をデバッグモードに遷移させるデバッグモード信号 1 0 3 をマスク信号として使用しても良い。

【 0 0 8 9 】

次に、本発明の第 3 の実施形態について説明する。

図 3 は、第 3 の実施形態によるリセット制御システムの構成例を示すブロック図である。なお、この図 3 において、図 1、図 2 に示したブロックと同じブロックには同一の符号を付し、重複する説明は省略する。

【 0 0 9 0 】

図 3 において、4 0' はプロセッサであり、D S U 4 1' と C P U コア部 1 2 とリセット制御部 4 2' とで構成されている。D S U 4 1' は、エミュレータ 6 0 とのインタフェースを有し、エミュレータ 6 0 から送信されるデバッグ用コマンド 1 0 1 を受信し、その実行結果 1 0 2 をエミュレータ 6 0 に送信する。D S U 4 1' は、エミュレータ 6 0 から送信されるデバッグ用コマンド 1 0 1 を受信すると、リセット選択信号 1 2 3 およびデバッグモード信号 1 0 3 を出力し、プロセッサ 4 0' をユーザモードからデバッグモードに遷移させる。また、受信したデバッグ用コマンド 1 0 1 をデコードする。

【 0 0 9 1 】

上記デコードの結果、エミュレータ 6 0 から受信したデバッグ用コマンド 1 0 1 がパワーオンリセットのリセットコマンドであったときには、パワーオンリセットに応じた第 1 のエミュレータリセット信号 1 1 7 を生成し、セクタ回路 7 1 に出力する。また、受信したデバッグ用コマンド 1 0 1 がハードリセットの

セットコマンドであったときには、ハードリセットに応じた第2のエミュレータリセット信号118を生成し、セクタ回路72に出力する。

【0092】

一方、デコードの結果、エミュレータ60から受信したデバッグ用コマンド101がリセットコマンド以外であったときには、デコード結果をデバッグ指示信号104としてCPUコア部12に供給する。CPUコア部12は、供給されたデバッグ指示信号104に従い処理を実行し、その実行結果105をDSU41'に通知する。さらに、DSU41'は、CPUコア部12から受信した実行結果105をエミュレータ60が送信したデバッグ用コマンド101の実行結果102としてエミュレータ60に送信する。

【0093】

セクタ回路71、72は、DSU41'から出力されるリセット選択信号123に基づき、外部のリセット発生回路等から入力されるパワーオンリセットに応じた第1の外部リセット信号115およびハードリセットに応じた第2の外部リセット信号116と、DSU41'から入力されるエミュレータリセット信号117、118のどちらか一方をシステムリセット信号120、121としてそれぞれ選択的に供給する。このセクタ回路71、72は、本発明のリセット選択手段を構成する。

【0094】

次に、図3に示すリセット制御システムの動作について説明する。

エミュレータ60からデバッグ用コマンド101が送信されておらず、プロセッサ40'がユーザモードで通常に動作しているとする。このとき、DSU41'からは第1のエミュレータリセット信号117および第2のエミュレータリセット信号118は出力されない。また、リセット選択信号123は外部リセット信号115、116を選択するように出力されている。

【0095】

このユーザモードにおいて、例えば、図示しない外部リセット発生回路等によりパワーオンリセットに応じた第1の外部リセット信号115が発生されると、発生された第1の外部リセット信号115はプロセッサ40'内のセクタ回路



7 1 に入力される。セクタ回路 7 1 では、DSU 4 1' から出力されているリセット選択信号 1 2 3 に従い、第 1 の外部リセット信号 1 1 5 を選択し、第 1 のシステムリセット信号 1 2 0 として出力する。

【0 0 9 6】

なお、ハードリセットに応じた第 2 の外部リセット信号 1 1 6 が入力された場合は、第 1 の外部リセット信号 1 1 5 が入力された場合とはリセット範囲が異なるのみで同様の動作である。

【0 0 9 7】

一方、エミュレータ 6 0 からデバッグ用コマンド 1 0 1 が送信され、プロセッサ 4 0' がデバッグモードで動作しているとき、DSU 4 1' から出力されるリセット選択信号 1 2 3 は、DSU 4 1' から出力されるエミュレータリセット信号 1 1 7、1 1 8 を選択するように出力されている。

【0 0 9 8】

このとき、エミュレータ 6 0 から送信されたデバッグ用コマンド 1 0 1 がパワーオンリセットあるいはハードリセットのリセットコマンドであったときには、DSU 4 1' は受信したリセットコマンドに対応した第 1 のエミュレータリセット信号 1 1 7 または第 2 のエミュレータリセット信号 1 1 8 の何れかの信号を出力する。

【0 0 9 9】

例えば、DSU 4 1' が、エミュレータ 6 0 からパワーオンリセットのリセットコマンドを受信すると、DSU 4 1' は第 1 のエミュレータリセット信号 1 1 7 をセクタ回路 7 1 に出力する。セクタ回路 7 1 では、DSU 4 1' から出力されているリセット選択信号 1 2 3 に従い、第 1 のエミュレータリセット信号 1 1 7 を選択し、第 1 のシステムリセット信号 1 2 0 として出力する。

【0 1 0 0】

ユーザモード時にセクタ回路 7 1 により第 1 のシステムリセット信号 1 2 0 として選択された外部リセット信号 1 1 5、またはデバッグモード時に第 1 のシステムリセット信号 1 2 0 として選択されたエミュレータリセット信号 1 1 7 は、分配されて、CPU コア部 1 2、コンパニオンチップ 5 0、カウンタ 4 6 に供

給される。

【0101】

CPUコア部12およびコンパニオンチップ50に供給された第1のシステムリセット信号120に従い、CPUコア部12およびコンパニオンチップ50が初期化される。また、カウンタ46は、供給された第1のシステムリセット信号120のネゲートを検出し、リセットネゲート検出から一定時間を経過した後、OR回路48に起動信号を出力する。

【0102】

OR回路48は、カウンタ46、47のそれぞれから出力される起動信号の論理和演算を行い、その結果を起動許可信号122としてCPUコア部12に供給する。この場合、カウンタ47から起動信号は出力されていないので、カウンタ46の出力がOR回路48からそのまま通過する。CPUコア部12はリセット処理の後、この起動許可信号122を受信して動作を開始し、リセットベクターから処理を実行する。

【0103】

また、第1のエミュレータリセット信号117の供給中に、何らかの原因で第1の外部リセット信号115が発生しても、リセット選択信号123に基づきセクタ71により第1のエミュレータリセット信号117が選択される。また、第1のエミュレータリセット信号117の供給中に第2の外部リセット信号116が発生しても、リセット選択信号123に基づきセクタ72により第2のエミュレータリセット信号118の側が選択されるが、このとき第2のエミュレータリセット信号118は出力されていないので、これに基づきリセットされることはない。

【0104】

また、プロセッサ40'がユーザモードで通常に動作しているとき、何らかの原因によりDSU41'からエミュレータリセット信号117または118が出力されたとしても、リセット選択信号123に基づきプロセッサ40'内セクタ71、72によりエミュレータリセット信号117、118は選択されないのので、出力されたエミュレータリセット信号117、118は無効となる。

【0105】

したがって、プロセッサ40'が通常に動作しているときには、DSU41'から出力されたエミュレータリセット信号117または118によって、CPUコア部12は内部状態を初期化されることなく通常動作を継続することができる。

【0106】

また、プロセッサ40'がデバッグモードで動作中に、エミュレータ60からリセットコマンド以外のデバッグ用コマンド101がDSU41'に送信され、送信されたデバッグ用コマンド101に基づき、CPUコア部12が処理を実行しているとする。このとき、仮に何らかの原因で、図示しない外部リセット発生回路等により外部リセット信号115または116が発生されても、リセット選択信号123に基づきプロセッサ40'内セクタ71、72により外部リセット信号115、116は選択されないで、出力された外部リセット信号115、116は無効となる。また、このときDSU41'からエミュレータリセット信号117、118も出力されていない。

【0107】

したがって、エミュレータ60からのリセットコマンド以外のデバッグ用コマンド101に基づいてCPUコア部12が処理を実行しているときには、図示しない外部リセット発生回路等により発生された外部リセット信号115または116によって、CPUコア部12は内部状態を初期化されることなく処理を継続することができる。

【0108】

以上のように、第3の実施形態によれば、エミュレータ60からのリセットコマンド101に応じてDSU41'から出力されるエミュレータリセット信号117、118と、外部のリセット発生回路等から入力される外部リセット信号115、116との何れかを、リセット選択信号123に基づいてセクタ回路71、72で選択し、システムリセット信号120、121として出力するようにする。

【0109】



すなわち、プロセッサ 4 0' がユーザモードで通常に動作しているときには、外部リセット信号 1 1 5、1 1 6 が有効となり、一方、プロセッサ 4 0 がデバッグモードで動作しているときにはエミュレータリセット信号 1 1 7、1 1 8 が有効となるようにする。

【0 1 1 0】

これにより、第 2 の実施形態で得られる効果に加え、デバッグを行っているときに不意に入力される外部リセット信号 1 1 5、1 1 6 によりデバッグ中の状態が初期化されるのを防ぐことができるだけでなく、通常動作時においても不意に入力されるエミュレータリセット信号 1 2 0、1 2 1 によって内部状態が初期化されるのを防ぐことができるようになる。

【0 1 1 1】

なお、本実施形態では DSU 4 1' がエミュレータ 6 0 からデバッグ用コマンド 1 0 1 を受信したとき、エミュレータリセット信号 1 1 7、1 1 8 を選択し、通常動作時には外部リセット信号 1 1 5、1 1 6 を選択するリセット選択信号 1 2 3 を生成しているが、DSU 4 1' がプロセッサ 4 0 をデバッグモードに遷移させるデバッグモード信号 1 0 3 をリセット選択信号として使用しても良い。

【0 1 1 2】

なお、本実施形態では外部リセット信号 1 1 5、1 1 6 かエミュレータリセット信号 1 1 7、1 1 8 かを選択するセクタ回路 7 1、7 2 を用いているが、図 2 に示した第 2 の実施形態において、マスク信号 1 1 9 を反転した信号に基づいてエミュレータリセット信号 1 1 7、1 1 8 をマスクするマスク回路を更に設けることにより、図 3 に示した第 3 の実施形態と同じ効果が得られる。

【0 1 1 3】

また、第 2 および第 3 の実施形態では、起動許可信号 1 2 2 を生成するカウンタ 4 6、4 7 をそれぞれのシステムリセット信号 1 2 0、1 2 1 に対して 2 つ備えているが、必ずしもこのカウンタはそれぞれのシステムリセット信号 1 2 0、1 2 1 に対して備える必要はなく、複数のシステムリセット信号に対して一つのみ備えても良い。このとき、複数のシステムリセット信号は、カウンタの前段で論理和演算を行いカウンタに入力する。

【0 1 1 4】

また、第2および第3の実施形態では、リセット動作後におけるプロセッサ40、40'の起動を一定時間だけ遅延するために、プロセッサ40、40'内にカウンタ46、47を設けているが、内部回路の作りによりプロセッサ40、40'よりコンパニオンチップ50の方が早くリセット解除となるような場合には、コンパニオンチップ50内にコンパニオンチップ50の起動を遅延するためのカウンタを設ける。プロセッサ40、40'とコンパニオンチップ50とでリセット解除までの時間差を調整する必要がある場合には、カウンタを設けなくても良いことは言うまでもない。

なお、カウンタの時定数はカウンタ毎に固定にしても良いし、カウンタの外部から設定可能にしても良い。時定数を設定する場合は、時定数をエミュレータからコマンドで与えても良いし、プロセッサ内部に初期化されないレジスタ等を設けて設定しても良い。

【0 1 1 5】

また、第2および第3の実施形態では、リセット制御部42、42'をプロセッサ40、40'内に設けているが、コンパニオンチップ50内にこれらと同様の機能を有するリセット制御部を設けても良い。

【0 1 1 6】

また、図1～図3に示した第1～第3の実施形態では、DSU11、41、41'をプロセッサ10、40、40'内に設けてエミュレータ30、60をプロセッサ10、40、40'に接続したが、DSU11、41、41'をコンパニオンチップ20、50内に設けてエミュレータ30、60をコンパニオンチップ20、50に接続しても良い。

【0 1 1 7】

また、図1～図3に示した第1～第3の実施形態では、外部リセット信号およびエミュレータリセット信号はそれぞれ2種類のリセット信号であるが、本発明は2種類のリセット信号からなるシステムのリセット制御システムに限られたものではない。

【0 1 1 8】



本発明の様々な形態をまとめると、以下のようになる。

(1) 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御システムであって、上記中央演算実行部の機能を別に実現するエミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号とに基づいてシステムリセット信号を生成して出力するシステムリセット出力手段を備え、上記システムリセット出力手段から出力される上記システムリセット信号を上記中央演算実行部および上記周辺制御部の両チップに供給することを特徴とするリセット制御システム。

(2) 上記システムリセット信号は、上記エミュレータリセット信号と上記外部リセット信号との論理和演算を行うことにより生成することを特徴とする上記(1)に記載のリセット制御システム。

【0119】

(3) 上記システムリセット出力手段は、上記周辺制御部のチップ内に備えられることを特徴とする上記(1)に記載のリセット制御システム。

(4) 上記システムリセット出力手段は、上記中央演算実行部のチップ内に備えられることを特徴とする上記(1)に記載のリセット制御システム。

(5) 上記エミュレータの動作時に上記外部リセット信号をマスクするマスク手段を備えることを特徴とする上記(1)に記載のリセット制御システム。

【0120】

(6) 上記中央演算実行部と上記周辺制御部のうち少なくとも一方のチップに、リセット後の起動タイミングを上記中央演算実行部と上記周辺制御部とで同期させる同期手段を備えることを特徴とする上記(1)に記載のリセット制御システム。

(7) 上記同期手段は、上記中央演算実行部あるいは上記周辺制御部のリセット後に、一定時間の経過後に起動許可を指示する起動許可信号を出力する起動許可信号出力手段により構成されることを特徴とする上記(6)に記載のリセット制御システム。

【0121】

(8) 上記中央演算実行部と上記周辺制御部のうち少なくとも一方のチップに

、リセット後の起動タイミングを上記中央演算実行部と上記周辺制御部とで同期させる同期手段を備えることを特徴とする上記（５）に記載のリセット制御システム。

（９）上記同期手段は、上記中央演算実行部あるいは上記周辺制御部のリセット後に、一定時間の経過後に起動許可を指示する起動許可信号を出力する起動許可信号出力手段により構成されることを特徴とする上記（８）に記載のリセット制御システム。

【 0 1 2 2 】

（１０）中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御システムであって、上記中央演算実行部の機能を別に実現するエミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号との何れかをシステムリセット信号として選択的に出力するリセット選択手段を備え、上記リセット選択手段から出力される上記システムリセット信号を上記中央演算実行部および上記周辺制御部の両チップに供給することを特徴とするリセット制御システム。

【 0 1 2 3 】

（１１）上記中央演算実行部と上記周辺制御部のうち少なくとも一方のチップに、リセット後の起動タイミングを上記中央演算実行部と上記周辺制御部とで同期させる同期手段を備えることを特徴とする上記（１０）に記載のリセット制御システム。

（１２）上記同期手段は、上記中央演算実行部あるいは上記周辺制御部のリセット後に、一定時間の経過後に起動許可を指示する起動許可信号を出力する起動許可信号出力手段により構成されることを特徴とする上記（１１）に記載のリセット制御システム。

【 0 1 2 4 】

（１３）中央演算実行部を備えるシステムのリセット制御システムであって、上記中央演算実行部の機能を別に実現するエミュレータの動作時に外部リセット信号をマスクするマスク手段を備えることを特徴とするリセット制御システム。

【0 1 2 5】

(1 4) 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御方法であって、中央演算実行部の機能を別に実現するエミュレータの動作時には外部リセット信号にマスクを施し、上記エミュレータのリセット指示に基づくエミュレータリセット信号とマスクの施された外部リセット信号とに基づいてシステムリセット信号を生成して、上記中央演算実行部と上記周辺制御部の両チップに供給することを特徴とするリセット制御方法。

(1 5) 上記システムリセット信号は、上記エミュレータリセット信号と上記マスクの施された外部リセット信号との論理和演算を行うことにより生成することを特徴とする上記(1 4)に記載のリセット制御方法。

【0 1 2 6】

(1 6) 中央演算実行部と周辺制御部とが別チップ上に形成されたシステムのリセット制御方法であって、中央演算実行部の機能を別に実現するエミュレータのリセット指示に基づくエミュレータリセット信号と外部リセット信号との何れかをシステムリセット信号として選択的に出力し、上記システムリセット信号を上記中央演算実行部と上記周辺制御部の両チップに供給することを特徴とするリセット制御方法。

【0 1 2 7】

【発明の効果】

以上説明したように、本発明によれば、エミュレータからのリセット指示に基づくエミュレータリセット信号と外部のリセット発生回路等からの外部リセット信号とに基づいてシステムリセット信号を生成して中央演算実行部と周辺制御部の両チップに分配出力するようにしたので、外部リセット信号に基づく初期化のときだけでなく、エミュレータからのリセット指示によっても上記中央演算部および上記周辺制御部の2チップともに初期化することができる。

【0 1 2 8】

また、本発明の他の特徴によれば、エミュレータを動作させているときには、マスク回路によって外部リセット信号をマスクするようにしたので、デバッグ中に不意に外部リセット信号が入力されてもその外部リセット信号を無効にするこ



とができ、中央演算実行部や周辺制御部は初期化されることなくデバッグ中の内部情報を保持することができる。

【図面の簡単な説明】

【図 1】

第 1 の実施形態によるリセット制御システムを適用したシステムの構成例を示すブロック図である。

【図 2】

第 2 の実施形態によるリセット制御システムを適用したシステムの構成例を示すブロック図である。

【図 3】

第 3 の実施形態によるリセット制御システムを適用したシステムの構成例を示すブロック図である。

【図 4】

エミュレータとプロセッサとからなる従来のシステムの構成例を示すブロック図である。

【図 5】

プロセッサとコンパニオンチップとからなる従来のシステムの構成例を示すブロック図である。

【図 6】

エミュレータとプロセッサとコンパニオンチップとからなる従来のシステムのデバッグ形態例を示すブロック図である。

【符号の説明】

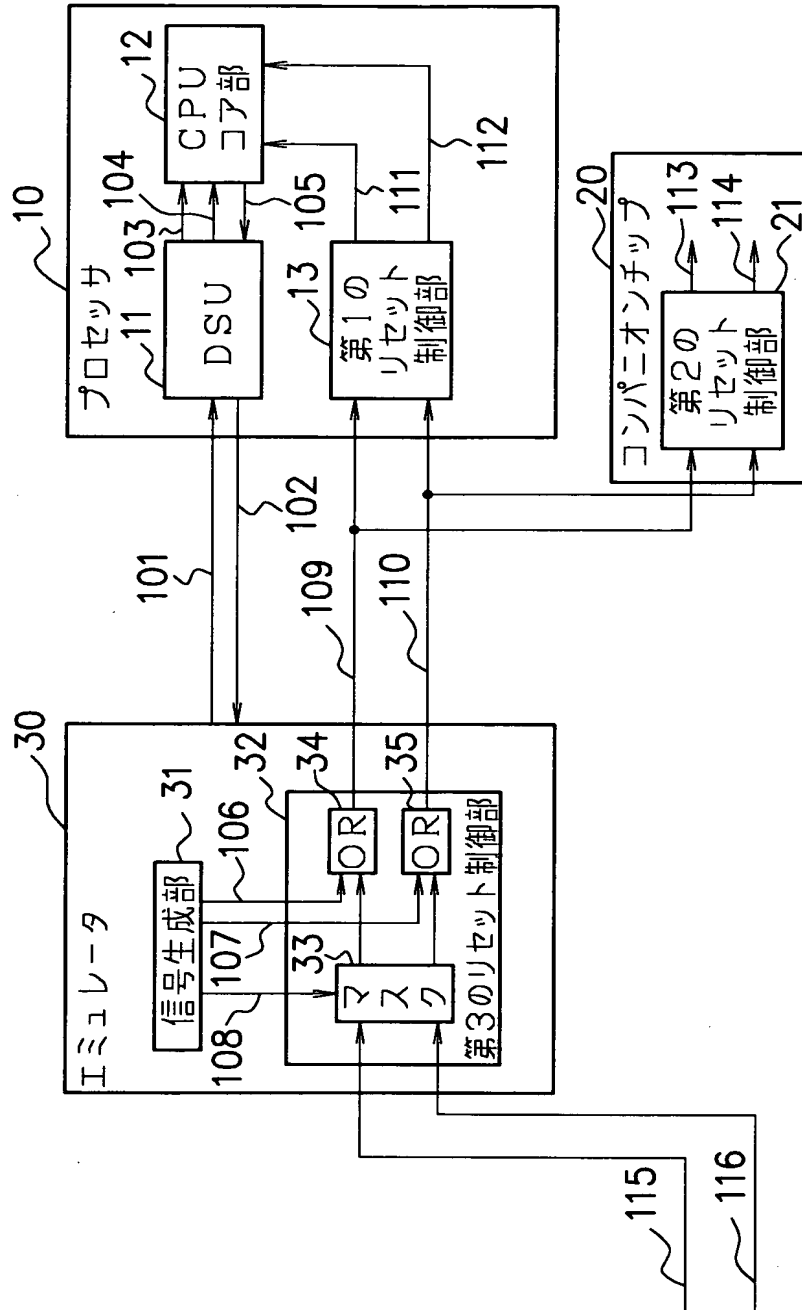
- 1 0、4 0、4 0' プロセッサ
- 1 1、4 1、4 1' D S U
- 1 2 C P U コア部
- 1 3 第 1 のリセット制御部
- 2 0、5 0 コンパニオンチップ
- 2 1 第 2 のリセット制御部
- 3 0、6 0 エミュレータ

3 1 信号生成部
3 2 第 3 のリセット制御部
3 3、4 3 マスク回路
3 4、3 5、4 4、4 5、4 8 O R 回路
4 2、4 2' リセット制御部
4 6、4 7 カウンタ
7 1、7 2 セレクタ回路
1 0 6、1 0 7、1 1 7、1 1 8 エミュレータリセット信号
1 0 9、1 1 0、1 2 0、1 2 1 システムリセット信号
1 1 5、1 1 6 外部リセット信号
1 2 2 起動許可信号

【書類名】

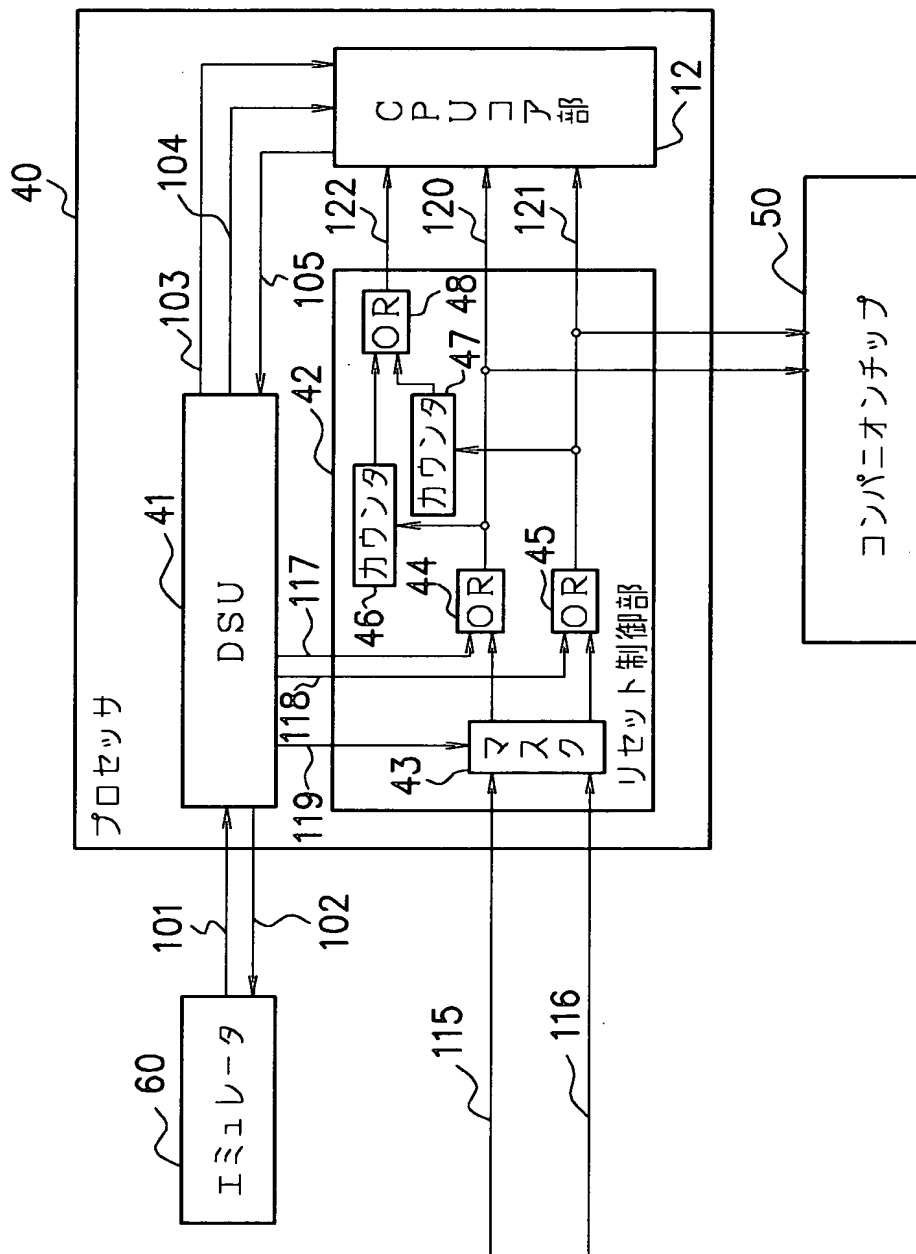
図面

【図 1】



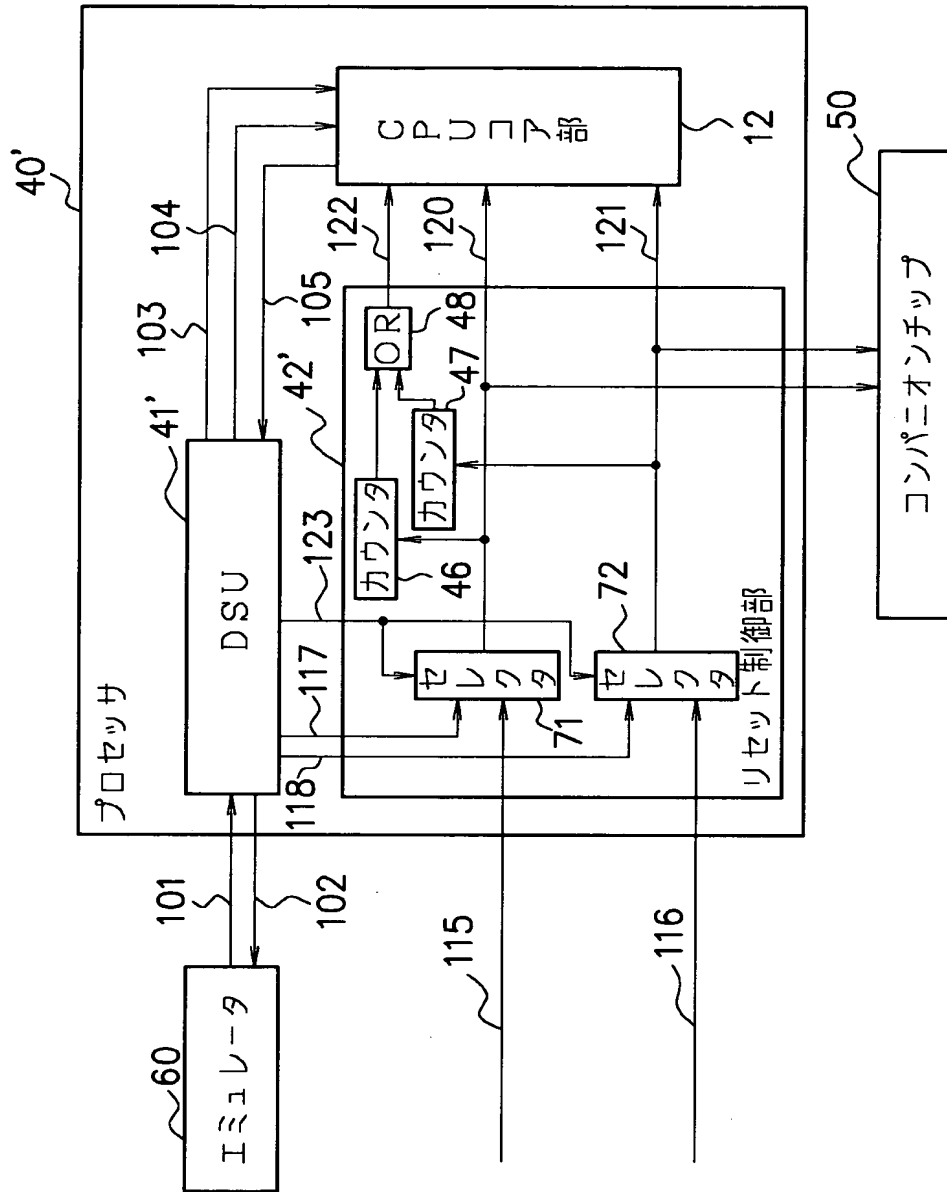
第 1 の実施形態

【図 2】



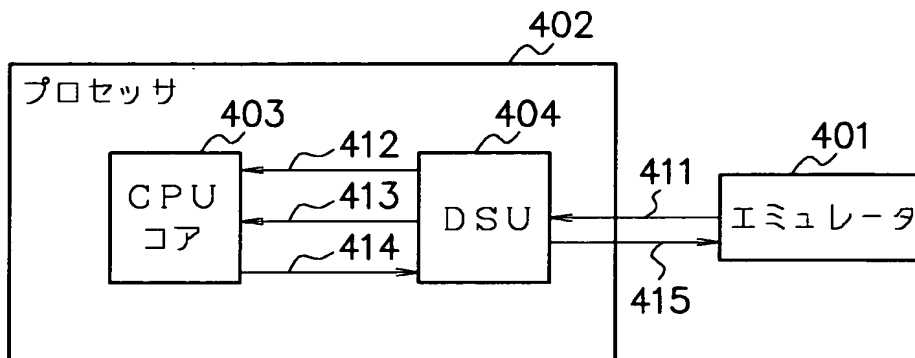
第2の実施形態

【図 3】



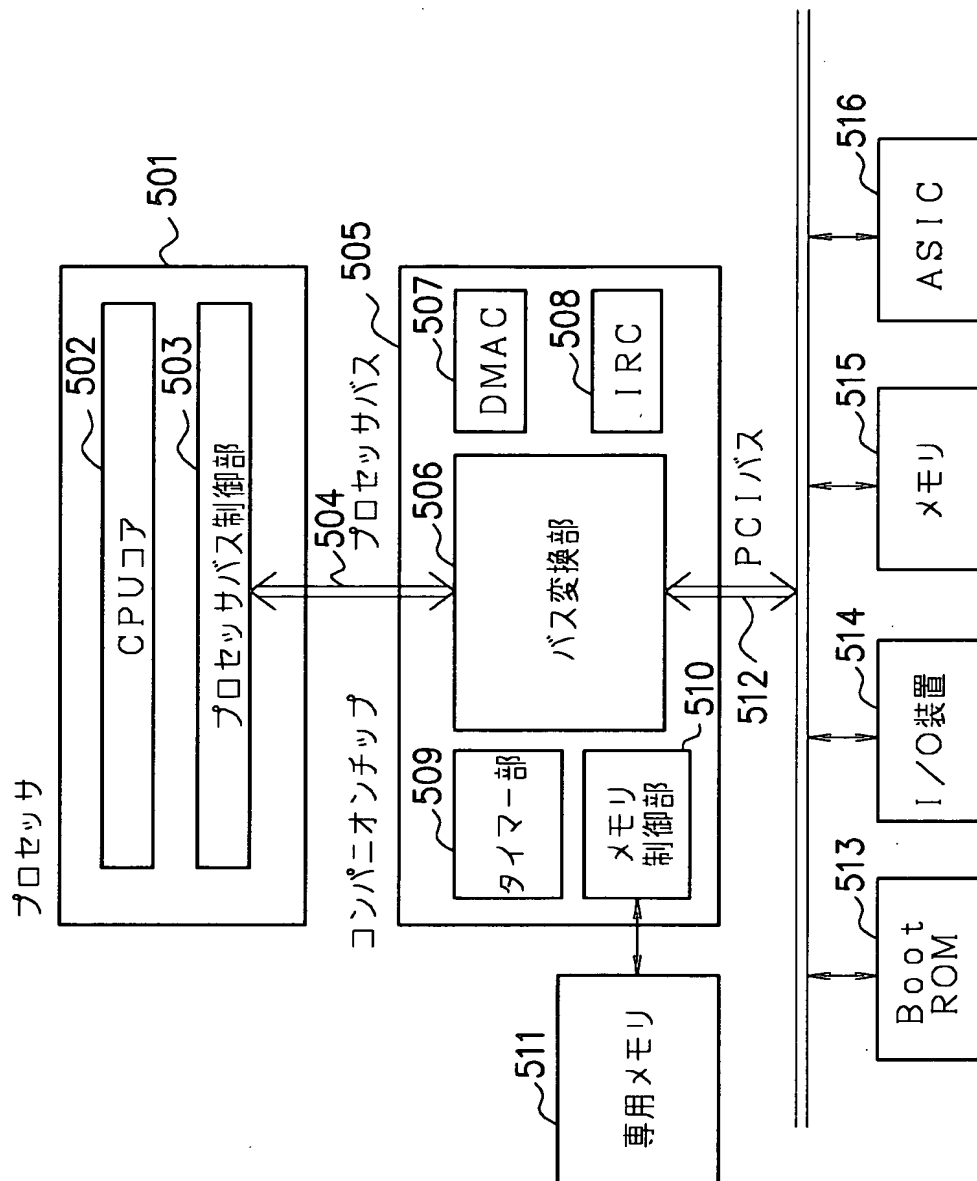
第3の実施形態

【図 4】



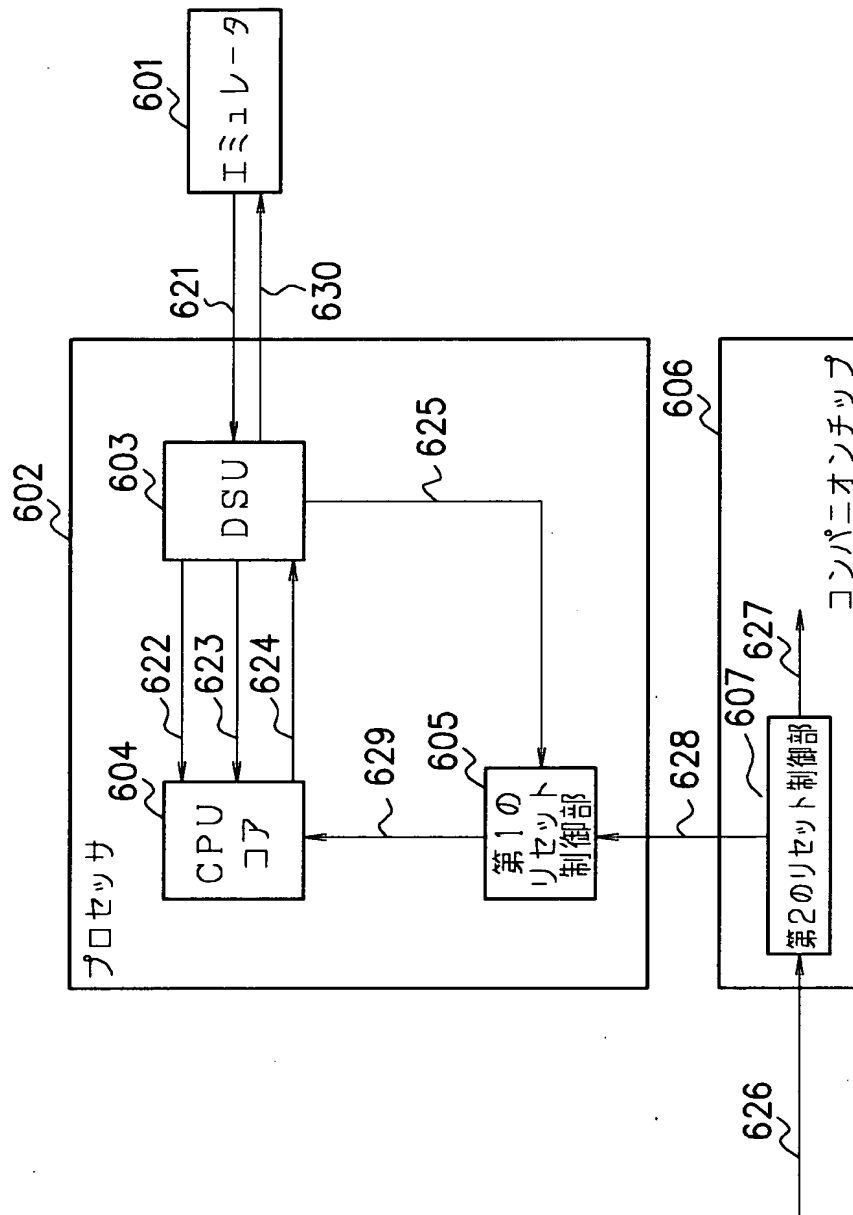
従来のシステムの一例

【図 5】



従来のシステムの他の例

【図 6】



従来のデバッグ形態例

【書類名】 要約書

【要約】

【課題】 エミュレータからのリセット指示により、プロセッサおよびコンパニオンチップの2チップともに初期化できるようにする。

【解決手段】 エミュレータ30のリセット指示に基づくエミュレータリセット信号106、107と、外部のリセット発生回路等から与えられる外部リセット信号115、116との論理和演算をOR回路34、35で行い、当該論理和演算の結果をシステムリセット信号109、110として、プロセッサ10とコンパニオンチップ20とに分配供給することにより、エミュレータ30からのリセット指示によりプロセッサ10およびコンパニオンチップ20の2チップともに初期化することができるようにする。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 2 3]

1. 変更年月日	1 9 9 6 年 3 月 2 6 日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号
氏 名	富士通株式会社